Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 484820, 10 pages
http://dx.doi.org/10.1155/2015/484820

Research Article

SensoTrust: Trustworthy Domains in Wireless Sensor Networks

Pedro Castillejo, José-Fernán Martínez-Ortega, Lourdes López, and José Antonio Sánchez Alcón

Next-Generation Networks and Services (GRYS) Research Group, Research Center on Software Technologies and Multimedia Systems for Sustainability (CITSEM), Universidad Politécnica de Madrid (UPM), Edificio La Arboleda, Campus Sur UPM, Ctra Valencia, Km 7, 28031 Madrid, Spain

Correspondence should be addressed to Pedro Castillejo; pedro.castillejo@upm.es

Received 1 August 2014; Revised 10 September 2014; Accepted 28 September 2014

Academic Editor: Muhammad Khurram Khan

Wireless sensor networks (WSNs) based on wearable devices are being used in a growing variety of applications, many of them with strict privacy requirements: medical, surveillance, e-Health, and so forth. Since private data is being shared (physiological measures, medical records, etc.), implementing security mechanisms in these networks has become a major challenge. The objective of deploying a trustworthy domain is achieving a nonspecific security mechanism that can be used in a plethora of network topologies and with heterogeneous application requirements. Another very important challenge is resilience. In fact, if a stand-alone and self-configuring WSN is required, an autosetup mechanism is necessary in order to maintain an acceptable level of service in the face of security issues or faulty hardware. This paper presents SensoTrust, a novel security model for WSN based on the definition of trustworthy domains, which is adaptable to a wide range of applications and scenarios where services are published as a way to distribute the acquired data. Security domains can be deployed as an add-on service to merge with any service already deployed, obtaining a new secured service.

1. Introduction

The utilization of wearable sensors to obtain useful human body parameters is nowadays a reality, partially due to the deployment of wireless sensor networks (WSNs) as the backbone of the new e-Health applications. A WSN is a network composed of small autonomous devices (sensor nodes or motes) that either incorporates sensors or includes the ability to incorporate them, providing capabilities for monitoring both individually and cooperatively physical or environmental conditions (such as temperature, light, or motion) in each of the possible locations of every node [1]. Some fields of application for these networks include environmental control, industrial control, automotive, medicine, defense and security industries, home automation, and so forth. Miniaturization enhancements provide a new generation of tiny sensors that can be embedded in wearable devices to provide helpful data for a wide range of applications.

A project to deploy a wireless sensor network must be able to deal with the challenges related to the intrinsic characteristics of these types of networks. One of the most important challenges is energy consumption. Wearable nodes operate autonomously (and are usually battery powered), so it is necessary for the processes and algorithms used to be efficient and have an energy-saving focus. Related to this, the aggregation of information through the network can be an interesting mechanism: some repeated data are not taken into account and not sent and, in this way, energy and resources are saved. Another challenge is the existence of different types of devices and platforms: there is no standardization in this kind of sensors nodes so it is desirable to abstract the hardware features by means of high-level abstract functions. This can be done with an intermediate level, called middleware [2].

Since these networks are being used in a wide variety of applications, security challenges must be addressed too. For example, in e-Health environments where biometric data and medical records are being shared, different national and international privacy laws are applied. As far as WSN security specific problems are concerned, we have concluded that

security in WSNs must support generic applications and the basic operation of the network, taking into account the following facts:

(i) these nodes are tiny devices,

(ii) they are energy- and resource-constrained regarding computational power, the amount of memory, radio bandwidth, and coverage,

(iii) there are no tamper-proof zones in the commercial hardware platforms to store sensitive information (passwords, keys, identifications, etc.),

(iv) due to exposure to adverse environmental conditions they tend to fail or lose their power and may be removed by unauthorized personnel,

(v) radio communications can be intercepted and cryptoanalyzed to extract data, keys, and so forth.

As a wireless system, security challenges are greater than in wired networks because of the open access to data transmissions. By using broadcast transmissions, other devices listening in the same frequency may intercept every communication between two nodes. Another WSN challenge is the possibility of a node being captured and cryptoanalyzed by a third party. If security keys, policies, and other cryptographic elements are accessed, a new spurious node could be introduced in the network, and several attacks can be deployed: denial-of-service, man-in-the-middle, data sniffing and/or data modification, routes spoofing, and so forth.

This paper is divided as follows. In Section 1 we introduce the emerging uses of wireless sensor networks, their network topology, node composition, and the challenges when working with them. This section also includes security issues addressed in these networks. In Section 2 a series of works related with security in WSNs is presented; additionally, the contribution of SensoTrust is described and compared to those proposals that already exist. The benefits when developing security protocols specific for WSNs are explained in Section 3. This section also includes how to deploy trust domains in wireless sensor networks. In Sections 4 and 5 it is presented how SensoTrust can be applied to e-Health scenarios and applications, as well as the validation processes in a real application. Section 6 includes the conclusions and future work to develop from this paper on.

2. Related Works

The very first proposal for securing emerging WSNs was a set of security protocols based on TinyOs, called SPINS [3]. SPINS provides data confidentiality, integrity, and authentication. TinySec [4] also provides the same security mechanisms; both proposals are based on symmetric cryptography without any key management system. Another example on specific TinyOs proposals is TinyKey [5] addressed with key management and included mechanisms for key generation, distribution, and rekeying.

A novel key predistribution scheme in WSNs is put forward by Subash and Divya [6]. As it can be guessed, key distribution is the main concept here. Their proposal is based on setting two links and two keys for each communication between two nodes. If one of the communication links is compromised, the other link is still available. The drawbacks are the great number of keys involved and the problems that may be faced when a node is captured (and thus all the keys contained are revealed). Our proposal reduces the number of keys needed and performs a rekeying process when a node is compromised.

SecFleck [7] proposes a platform-specific security add-on, based on an Atmel TPM (trusted platform module) chip. Although this solution provides a fast and energy efficient way to support both symmetric and public key cryptography (including a tamper-proof chip for storing keys), it is a platform-specific solution, since it has been designed and validated only in Fleck nodes. SensoTrust is a platform-independent solution and can be deployed either as a standalone or composed service. Furthermore, a large number of proposals have been published involving authentication schemas. In [8] Khan and Zhang put forward an authentication schema based on smart cards and fingerprints. In addition to that, Yoon et al. [9] perform the cryptanalysis of an authentication scheme.

A general overview of trust systems and a simulation-based study is presented in [10], where Karthik et al. select five trust models and evaluate them using TRMSim. Results present BTRM_WSN as the best-case trust model.

TRMSim-WSN [11] is a Java-based trust and reputation model simulator aimed at providing an easy way to test a trust and/or reputation model over WSNs, along with comparing it against other models. Another trust calculation schema is presented by Karthik and Dhulipala in [12].

The first idea of a decentralized trust system was presented by Blaze et al. [13]. They proposed policymaker, a unified decentralized trust management system based on a simple language for describing security policies, credentials, and relationships.

A trust system for WSNs is presented by Boukerch et al. in [14]. They provide an agent-based trust and reputation management scheme (ATRM) for wireless sensor networks, assuming that mobile agents are resilient against the unauthorized analysis and modification of their computation logic.

In [15], Oleshchuk demonstrates how the concept of trust can be used to increase security in wireless sensor networks without using cryptography. It is done so by taking into consideration trustworthiness of individual sensors and monitoring each sensor activities. Plus, Dhulipala et al. present a specific WSN trust system in [16].

A particular trusted routing protocol for MANETS is presented in [17]. Abusalah et al. propose a trust-aware routing protocol (TARP) for secure-trusted routing in mobile ad hoc networks. In TARP, security is inherently built into the routing protocol where each node evaluates the trust level of its neighbors based on a set of attributes and determines the route taking into account these attributes. Recently, there have been several contributions to trust routing in WSNs [18–22].

SensoTrust, as described in the following sections, is a trustworthy domains model used to deploy security services in a WSN. It includes not only a complete trusting scheme to

accept, control, and exclude nodes participating in a trusted domain, but also mechanisms to define security policies using symmetric and asymmetric cryptography. Considering the WSN nodes constrains, security schemas have been designed to reduce the cryptographic material stored in each node. It also minimizes the data loss suffered when one of the nodes is compromised. In order to achieve the resilience goal, mechanisms for reconfiguring a compromised trusted domain (with rekeying or server reassignment) have been defined. These mechanisms are important for applications with two severe requirements: data security and quality of service.

3. Deploying Trust Domains to Increase WSN Security

As WSNs are being widely used in many applications (some of them with severe requisites such as critical infrastructures monitoring, defense, surveillance, and e-Health), including security mechanisms is of major importance. Due to node limited capabilities, traditional network security schemas are not suitable, so new schemas must be designed.

Privacy is an important feature to care about. If WSNs are utilized to share private data (medical records in hospital monitoring, timetables in access control, location in tracking systems, etc.), some mechanisms must be implemented to protect private information from unauthorized access.

Access control is also an important mechanism. In order to protect data in the network, some control steps must be applied to give access only to authorized nodes. To gain access, nodes must be authenticated inside the network.

Finally, if we want to ensure that the data received by a node has not been modified in the route taken from the sender, an integrity control mechanism is necessary.

All of these mechanisms can be addressed in a trustworthy domains environment, where any new node joining the network must authenticate itself to obtain the necessary keys to start sending and receiving data. When nodes are interconnected, several surveillance policies must be applied in order to exclude a suspicious node from the network, preventing the node from participating in it anymore. The aim of deploying a trust domain is achieving a platform-agnostic security mechanism that can be used regardless of network topology or application requirements. Resilience is a very important concern too. In fact, if a stand-alone and self-configuring WSN is needed, an autosetup and a reconfiguring mechanism are required as well, so as to maintain an acceptable level of service in the face of security issues.

3.1. Trust Definition and Trust Domains. Trust implies reliance on another person or entity. In the field of knowledge related to common networks, trust can be determined by the reliance that each one of the nodes has on the others. Thus, it is very important to measure this feature in some kind of way. In WSN, trust increases its importance. Since it is very easy to introduce a new node in a deployed WSN and capture and analyze the data traffic, some mechanisms must be developed in order to measure and assure that all the nodes participating in a WSN can rely on each other. In our proposal, several mechanisms are included: trust domains, polling systems, and measures to undertake when trust has been lost in a domain. Overall, steps are needed in order to have a functional trustworthy mechanism.

The first step is trust establishment: a new node participating in the network must publish its authorization to start communications with other nodes. Once the new node has been granted access, trust is propagated through the network.

3.2. Validating Trust Domains and Polling System. When a trust domain has been defined, and all the elements needed have been deployed (keys, roles, policies, etc.), nodes are able to start registering services and sending data. Validation of the trust domain must ensure that all the nodes participating in each domain can reach other nodes and have obtained both the keys and the security policies. Trust must be measured in some way to determine if one trusted node has been compromised or has an unexpected behavior (in case a retrusting mechanism has to be started). In our proposal, trust is measured by a polling system.

Once the domain has been validated, and the trust in that domain can be assured, the network will start its usual working. As an expandable network, new nodes can be inserted within the ecosystem to extend the range or the services offered by the network itself.

The way to assure that new nodes behavior does not compromise the prior trust level is by watching the new nodes actions and comparing them with the policies defined. If we distribute this surveillance system across the network, each node can report to the head-node (broker, security manager, or cluster head node) the actions taken by the other nodes. In this way, compromising the network due to new nodes, or a trusted node that changes its behavior to a risky one, can be avoided.

The polling system is simple and effective: each node assigns votes for every neighbor in its range. In a WSN, each node can promiscuously read the data sent by its neighbors, because communications are broadcasted. Data sent means messages are being generated by each node. If these messages do not match the rules defined in the security policy, a negative vote is emitted. The security manager should count votes for each node, and if the negative result exceeds a predefined threshold for a node, some actions might be taken: waiting for predefined time (quarantine), excluding the node from the network, redefining a new trust domain, and so forth.

3.3. Actions to Take When Losing Trust in a Domain. When one of the nodes participating in a trust domain becomes compromised, the whole domain should be treated as compromised. Several actions must be taken in order to reestablish the trust in that domain. If the compromise affects a regular node, it is enough to renew the key that protects communications within that domain and then redistribute the new key to all the nodes in the domain but the compromised one. If the compromised node is the domain server (or cluster head or broker, as previously named), it will be also necessary

to designate a new domain server and, right afterwards, renew the key.

In the next sections the deployment model to evaluate our proposal will be described, including all the mechanisms to define, evaluate, and redefine a trust domain.

4. Description of the SensoTrust Proposal

In previous paragraphs we have proposed mechanisms to define security domains and establish trust in WSNs. In order to address all the characteristics from the proposal, we have also designed an implementation model to deal with all the security issues regarding security domains. SensoTrust (the proposed model based on previous section directions) is composed by several elements, which are presented below.

The *security manager* (SM) is the principal actor of the model and should be an external host able to generate symmetric and asymmetric keys in a secure manner. *Several nodes* (N) sharing the same *domain key* (DK) and some common aspects (mission, localization, capabilities, etc.) define a *trust domain* (TD). Each TD has a *trust policy* (TP), including the behaviors, measures, actions, and other features regarding the life-cycle of a trust domain. Also, a *domain key server* (DKS) exists in each TD. It is a special node in charge of distributing the domain key when needed. A single node can act as a DKS if the security manager decides it. Any node in the network has its own *node ID* (NID), unique identification, and an individual node key (NK). The ciphering methods and key sizes used when ciphering a message are defined inside the *ciphering suite* (CS). Two types of keys are used in public key infrastructure (PKI) cryptography: *public key* (Kp), the part of the key assumed to be freely distributed, and *Private Key* (Ks), the part of the key to be kept in secret and not distributed.

Since WSN nodes have several limitations, a full and robust public key infrastructure is unlikely to fit in these nodes. Due to this reason, a hybrid PKI and symmetric keys schema has been used in this proposal. PKI is used for the communication between special nodes (security manager and domain key server), and a simple-but-secure AES symmetric key for the plain nodes. The size of this AES key is constricted by the nodes computing power and, in order to have a scalable architecture, it becomes defined by a ciphering suite field in the ciphered data packets, so as to determine the length of the key or select a new algorithm (such as RC5, etc.).

4.1. Example Scenario for an e-Health Application. In the previous section, SensoTrust elements were presented. As a way to describe in depth the role performed by each of them, the SensoTrust schema has been illustrated with an application example: a medical environment with sensitive data interchange (medical records, biometric parameters of the patients, medical stuff accessing privileges, etc.). This application example deployment is shown in Figure 1.

In this basic example, only two trust domains have been defined (as there are two levels of security): a trust domain for the admission area and another trust domain for a postsurgery room. At the admission area, the information circulating is related to each patient medical records and extra information (contact information, personal id, etc.). In this trust domain, the security policy must guarantee, at least, information privacy.

At the postsurgery area the information in the network is composed of medical records and real-time biometric parameters (body temperature, blood pressure, ECG, breathing rate, etc.). In this trust domain, not only privacy but also data integrity must be assured. Furthermore, routing and data delivery are important when an alarm is issued (e.g., with a low breathing rate). In the next paragraphs, each SensoTrust element is identified and described.

The security manager is an external entity (but connected through the sink to the WSN), where high level security mechanisms have been deployed. The tasks assigned to the security manager include node key generation and distribution, security policies definition and distribution, domain key server role assignment, rekeying processes, and data recording (with regards to the nodes participating in each trust domain).

Trust domains definition can be done following several policies: the number of nodes per domain, node localization, domains for specific functionalities (e.g., temperature measurement, intrusion detection, environmental monitoring, etc.), or any other distribution demanded by a specific application.

Focusing on a generic trust domain (illustrated in Figure 2), the main components and elements will be described.

For each trust domain defined, a domain key server is assigned (either in deployment or runtime) and a domain key is shared. Any node participating in a domain can assume the DKS role if the former DKS has been compromised. If the compromise only affects a regular node, a rekeying method is enough: DKS requests SM a new DK, and then this new key is sent to all nodes but the compromised using each node key.

Domain key server stores its own pair of public and secret keys (along with the public key of the security manager, so as to communicate with it) and the domain key to communicate with nodes participating in the domain and the security policy for the domain.

The other nodes also need to keep some important information: the node ID and key, the domain Key, the public key of the security manager (used if security manager assigns the role of the former domain key server to a new node), and also the domain security policy, where the surveillance parameters and polling system are described.

Cryptographic information shared by nodes has been reduced to a minimum in order to minimize the risk when a node is compromised. In our schema, if a single node or a DKS is compromised, only the domain key becomes evident (as the public key of the SM can be revealed without security issues). Redistributing a new DK or designating a new DKS will be enough to recover the trust.

4.2. Federation of Critical Nodes in the Network for Resilience Improvement. There are some special nodes in WSNs that are of critical importance for the satisfactory performance of the network. Depending on the network topology, these special nodes can be named as cluster heads, relays, brokers, and so

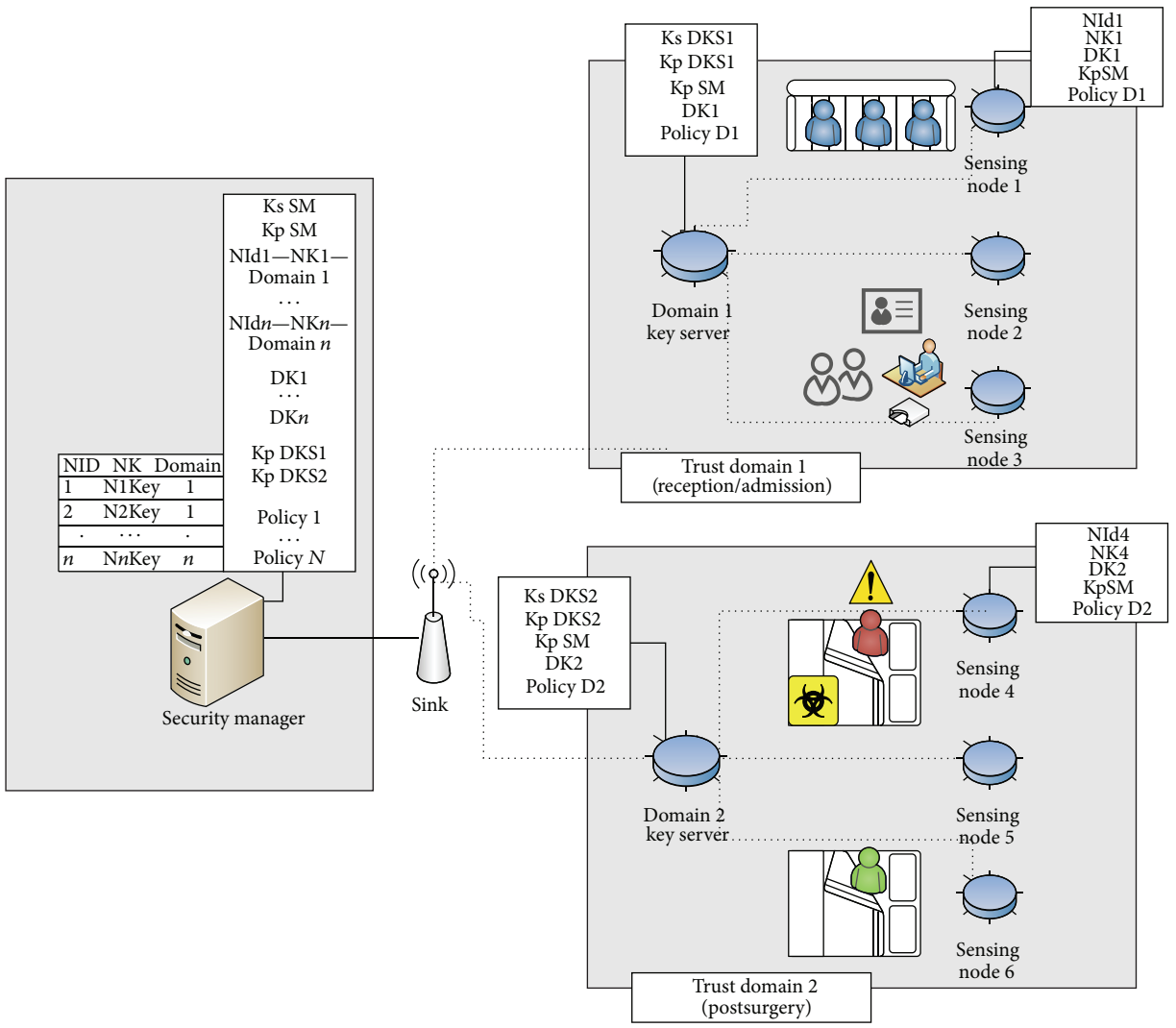

Figure 1: Proposed model applied in medical environments for an e-Health application.

forth. An efficient solution is implementing a mechanism to replace one of these nodes if they get compromised or stop working properly (due to low battery, falling out of coverage, etc.) as a method to improve the resilience of the network.

A good example of this solution is broker federation. In a network with some nodes acting as brokers to deploy and publish the services, should the broker be compromised, the applications are not able to access sensor data anymore. The solution to this issue is reassigning the role of "broker" to another node. This reassignment is based on node reputation (using the polling system), node battery status, or node capabilities. From this moment on, the new broker will publish all the services to the applications and the network will work properly again. SensoTrust includes this possibility, since the domain key server can be considered as the broker or the cluster head, and DKS reassignment mechanism has been developed in our proposal. When a new node starts acting as a DKS/cluster head/broker, the former one can be suspended if it is considered as a spurious node. The complete process is presented in the next section.

4.3. Sequence Diagrams for Processes. Several messages have been defined to model the communication between the elements of the system. There are messages to join a domain (HELLO), renew the domain key, query and send a node key, assign a new domain key server when the active one has been compromised, and renew the compromised keys. In next paragraphs, sequence diagrams describing these messages are presented.

The first step that a new node has to complete when entering a WSN is joining a defined trust domain. A HELLO message is sent by the new node (Node1 in Figure 3). Data in this message includes a random number (x) ciphered with the node key (only DKSs having that key will be able to decipher it), the ciphering suite used, and its unique node ID. This message is broadcasted to one (or several) active domain key servers, and the DKSs will prompt the security manager for the Node1 key (using node ID received as an index) to be able to cipher and decipher data to/from Node1.

The SM will decide which one of the querying DKS will be answered (the SD where the new node should join).

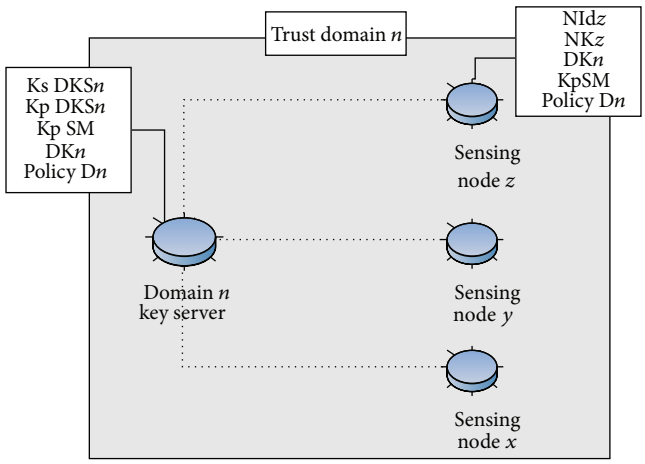

Figure 2: Generic trust domain components.

The DKS designed will receive the Node1 key and will use it to cipher the data included in the HELLO REPLY message. These ciphered data include the random number sent by Node1 (in order to confirm to Node1 that the responding DKS is a trusted one) and the key for domain participation. Additionally, the Node ID is added.

When Node1 receives the HELLO REPLY message, it will use its own Node key to decipher it obtaining: the random number (x) previously sent in the HELLO message (checking that the replying DKS is a trusted one, since it has been able to decipher the HELLO message) and the domain key for the trust domain assigned.

At this point, Node1 has access granted to trust domain 1, since it has obtained domain 1 key and is able, for example, to publish its services to the network (ciphered with the Domain 1 Key) and then they get published by a service orchestrator or broker existing in the WSN.

Another interesting process is the ability of the system to renew a domain key that has been compromised (e.g., if a node is captured and cryptoanalyzed). This process, described in Figure 4, starts with a DKRenew message sent by the DKS to the SM containing the compromised key (ciphered with the SM public key—Kp—guaranteeing than just the SM will be able to open it), ciphering suite used, and DKS1 identification. The SM will respond with a message containing the new domain key (ciphered with DKS1 public key), ciphering suite used, and DKS1 identification. In order to reobtain Domain 1 trust, the new domain key must be sent to all the nodes but the compromised one. A DKRenew message is used, which contains the new DK ciphered with the individual node keys (one message per node).

The worst case is when the compromised node is the domain key server (which can also be named as the cluster head or the broker as presented in Section 4.2). The security manager will choose a new node to assign the role of DKS. This election can be done using several variables, defined in the security policy: node reputation (using the polling system), node battery status, node capabilities, and so forth.

This process starts, as described in Figure 5, with a DKS_ASSIGN message using double ciphering. First, all the cryptographic information needed by new DKS to start working is added: the new domain key, a public/private key pair for this new DKS, and a random number (x). This data is ciphered using the SM secret key. Then, another ciphering is done with the new DKS symmetric key. This double ciphering guarantees two aspects: the only node able to decipher the message will be the new DKS (since it is ciphered with its symmetric key) and this new DKS will be able to confirm that the message was generated by the trusted SM, using the SM public key to decipher the message previously signed by the SM.

SM needs confirmation from the new DKS, so a reply message is sent. Again, a double ciphering is used. First, the random number (x) sent by the SM is ciphered with the DKS secret key, and then the message is ciphered with the SM public key, guaranteeing that just the SM will be able to decipher the message and will confirm that it was generated by the assigned DKS and not by a spurious one.

Once the new DKS is assigned and confirmed, the new domain key will be sent to all the nodes but the compromised DKS, as described in Figure 4 (domain key renewal).

Table 1: Security requirements and solutions.

Requirement	Solution
Authentication	Node ID, PKI
Availability	Node revocation
Privacy	Keys, PKI
Data integrity	Hash
Prot. outsider attacks	Node ID, PKI
Prot. insider attacks	Security policies, trust domains

4.4. Security Analysis. The presented proposal is able to provide solutions to cover a wide range of security requirements. They fulfill the majority of the security requirements for WSNs-based applications, from the less critical ones (animal control, traffic density measurement, or environmental monitoring) to the most security-sensitive requirements needed by e-Health applications, which demand data integrity, privacy, authentication, and so forth. For each security requirement identified, a solution has been included in SensoTrust, as depicted in Table 1.

Furthermore, the security proposal has been evaluated against a well-known group of security attacks selected from the literature [23, 24]. For each attack, the countermeasure implemented in SensoTrust proposal is described as follows.

(i) *Denial of service (DoS)*: this is the least sophisticated attack. It appears when either the physical layer is degraded to a level where the communication between nodes is impossible (jamming) or when a spurious node starts sending malicious data packets to the network. In both situations, an alarm is triggered in the SensoTrust security manager to alert the IT-security staff.

(ii) *Sybil attack* occurs when a node is asking for multiple IDs. If the attack succeeds, the node is able to subvert the trust mechanism. In SensoTrust, every node ID

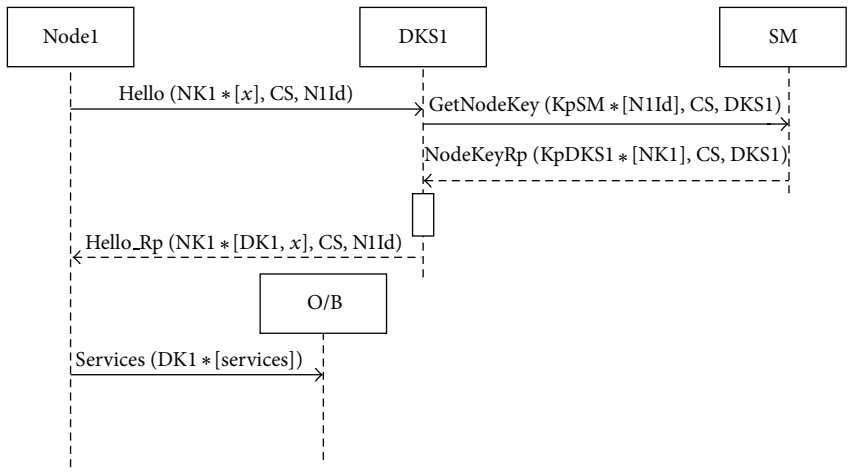

Figure 3: Sequence diagram describing a new node registration into a trust domain.

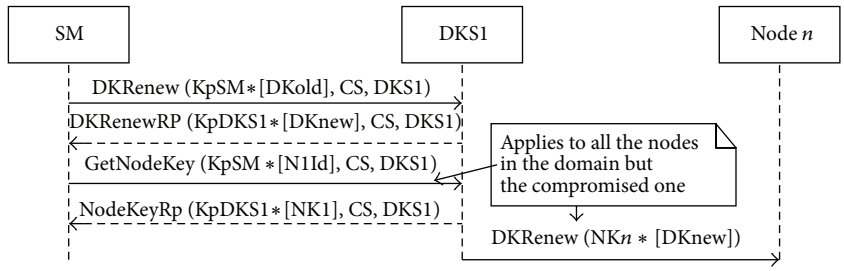

Figure 4: Sequence diagram describing how the security manager starts the renovation of a domain key.

is preconfigured for each node and only the security manager (out of the WSN) has the complete list of the IDs. Furthermore, it is possible to perform a node revocation (as explained in the previous section).

(iii) *Message corruption*: in this attack, a message reaches the recipient with a different content than the one sent by the source. This situation is either because the message has been degraded in the transmission, or because the message has been intercepted and intentionally changed. To avoid both issues, SensoTrust includes the ciphering suite functionality, which allows performing a message hash (using MD5, SHA1, etc.).

(iv) *Eavesdropping*: WSNs use broadcast transmissions, so other devices listening in the same frequency may intercept every communication between two nodes. To avoid data disclosure, SensoTrust provides both symmetric and PKI ciphering capabilities.

(v) *Node subversion*: if one of the nodes is captured and cryptoanalyzed the secret keys, node ID, security policies, and so forth are disclosed. SensoTrust aim is to minimize the cryptographic and security information stored in each node. Nevertheless, if a node is captured, all the keys in the network can be renewed as explained in previous section.

(vi) *Node replication* occurs when a node ID is copied, replicated in a new node, and then introduced in the network. From that moment on, the network accepts the node with the cloned ID as an authorized node. SensoTrust provides two mechanisms to avoid this attack. The first one is the Node ID, which is stored in an external entity (the security manager) that controls all the IDs working in the network. The second mechanism is security policy. If the security manager detects that 2 nodes are operating with the same ID, a node revocation protocol is issued, and the node is dropped from the network.

(vii) *False node*: this attack introduces data traffic in the network to avoid legitimate nodes to communicate (injecting false data messages, claiming for authorization continuously, etc.). Using the node ID, SensoTrust is able to identify the false node and, using the domain key renewal functionality, all the messages sent by this node will be discarded.

Attacks and countermeasures have been summarized in Table 2.

5. SensoTrust Validation in a Real Application: Results

We needed to validate our security proposal against a complete, deployed platform in an e-Health scenario, so it was

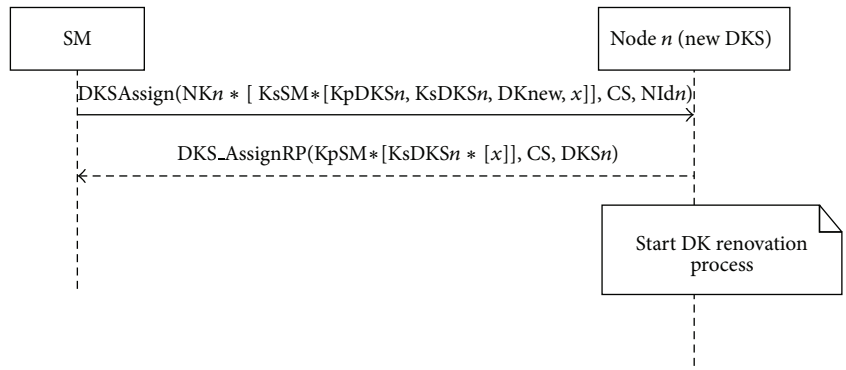

Figure 5: Sequence diagram describing how the security manager designates a new domain key server.

Table 2: Attacks and countermeasures.

Attack	Countermeasure
DoS	SM alarm
Sybil attack	Node ID, node revocation
Message corruption	Hash
Eavesdropping	Keys (symmetric and PKI)
Node subversion	Few crypto. data stored
Node replication	Node ID, security policies
False node	Node ID, domain Keys

decided to use the one developed for the project "AWARE: Accessible Wearable Device Platform for Smart Environments," as the research group the authors work on is involved in the development of this project. The main goal of AWARE is to define a ubiquitous and accessible platform for the deployment of services in intelligent environments through the use of wearable devices. One of the deployments is based on the extension of the middleware developed within the LifeWear project [25]. LifeWear intends to improve the quality of human life by using wearable equipment and applications for everyday use. The main objective of LifeWear is the development of modern physiological monitoring to inspect human health status in different environments, so that actions and safety critical issues will be real-time monitored. This means, for example, that blood pressure, pulse, or body temperature of a patient can be tracked with wearable devices and sent to the medical staff at a fast pace to control the correct fulfillment of a treatment. It is also possible to use mobile technology to build computer-based online services for people, such as virtual training environments, or monitor workers in hostile work environments that demand a high state of alertness, such as firemen. This means that wearable-enabled online monitoring of human bodily states has a wide range of application possibilities, as long as the related issues can be solved.

One of these related problems is security. Since medical parameters are sent (blood pressure, temperature, pulse, etc.), data privacy is a must. A security mechanism to provide data confidentially and integrity can be deployed with our proposal (SensoTrust), integrated in the system as a new service that can be combined with already deployed ones.

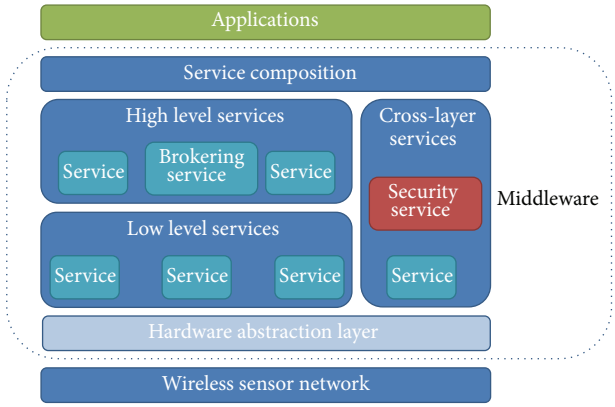

Figure 6: AWARE platform, including LifeWear's middleware and security component.

The AWARE architecture, including the middleware proposed in LifeWear, is displayed in Figure 6. The middleware is composed by four abstraction layers, regarding the functionalities covered in each of them, namely, hardware abstraction layer, low and high services, cross layer services, and service composition platform.

The hardware abstraction layer includes the sensor node hardware platform, the operating system, and the networking stack. It offers an easy way to port the solution to other hardware platforms.

The low and high services layers define the software components needed to abstract the underlying network heterogeneity, thus providing an integrated, distributed environment to simplify the programming tasks by means of a set of generic services, along with an access point to the management functions of the sensor network services.

The upper layer is the service composition platform, designed to build applications using services offered by the lower layers.

The cross-layer services are offered to both high and low level services in order to provide inner service composition. The security system presented in this paper has been deployed as a service inside this layer. The security service can be used by the upper layer (service composition) to compose new

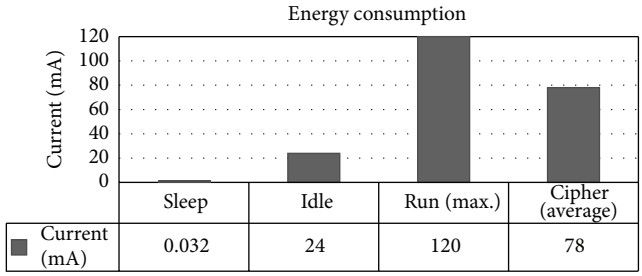

Figure 7: Energy consumption per process.

Table 3: Delay introduced by the security service.

Process	Time (ms)
New node authentication	739
New service publication	525

secured services, based on the services presented in the lower layers.

The architecture has been deployed over a commercial WSN node solution: SunSpot platform, manufactured by Oracle. Main characteristics of SunSpot hardware (rev. 6) platform are as follows:

(i) processor: ARM 920T CPU (180 MHz-32 bit),

(ii) memory: 512 Kb RAM, 4 Mb FLASH

(iii) network: Chipcon 2420 radio with integrated antenna (IEEE 802.15.4 at 2.4 GHz),

(iv) Data: USB interface—mini-b connector,

(v) power supply: 3.6 V rechargeable 750 mAh Li-Ion battery.

As mentioned above, a new security service has been deployed over this platform, and several tests were carried out to obtain data about the behavior of the system. Since our proposal only requires cryptography calculation when it is needed, energy efficiency is assured. Only messages of critical importance will be ciphered, reducing power consumption in a large number of processes that do not require it. Energy consumption of different processes is shown in Figure 7.

Delay introduced by the system is also another important parameter to measure. Delay introduced by the security service is shown in Table 3. Two basic operations have been measured: the authentication of a new node in the network and the publication of a new service when a node has been already authenticated (including ciphering delay).

6. Conclusions and Future Work

In this paper, a trustworthy domains model (SensoTrust) to deploy security services in a WSN has been presented. Furthermore, we have proposed a complete trusting scheme to accept, control, and exclude nodes participating in a trusted domain, defining security policies, and using symmetric and PKI cryptography and special elements to achieve that goal (security manager and domain key servers). Also, resilience issues have been addressed, proposing mechanisms to reconfigure a compromised trusted domain with rekeying or servers reassignment. Security schemas have been designed to reduce the cryptographic material stored in each node in order to minimize the data loss suffered when one of the nodes is compromised. The proposal has been tested and validated in a real e-Health application: the AWARE project.

An open issue is checking the influence of our system as for power consumption (battery life) in a wide range of WSN commercial solutions. Several benchmarking tests should be performed to obtain measures and comparatives.

Finally, testing our proposal in different applications and scenarios will give us important data about modifications and new security schemas.

Conflict of Interests

The authors declare that they have no conflict of interests to disclose.

Acknowledgments

The work presented in this paper has been partially funded by the Spanish Ministry of Economy and Competitiveness in the framework of the Project "AWARE" (TEC2011-28397). The authors would like to thank CITSEM Research Center from the UPM.

References

[1] M. Galetzka, J. Haufe, M. Lindig, U. Eichler, and P. Schneider, "Challenges of simulating robust wireless sensor network applications in building automation environments," in *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA '10)*, pp. 1–8, Bilbao, Spain, September 2010.

[2] Y. Ning, R. Wang, S. Ma, and Z. Wang, "An application of context middleware based on fuzzy logic for wireless sensor networks," *Wireless Sensor Network*, vol. 1, no. 5, pp. 365–369, 2009.

[3] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[4] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, ACM, New York, NY, USA, November 2004.

[5] R. D. Corin, G. Russello, and E. Salvadori, "TinyKey: a lightweight architecture for wireless sensor networks securing real-world applications," in *Proceedings of the 8th International Conference on Wireless On-Demand Network Systems and Services (WONS '11)*, pp. 68–75, January 2011.

[6] T. D. Subash and C. Divya, "Novel key pre-distribution scheme in wireless sensor network," in *Proceedings of the International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT '11)*, pp. 959–963, Tamil Nadu, India, March 2011.

[7] W. Hu, P. Corke, C. Shih, and L. Overs, "secFleck: a public key technology platform for wireless sensor networks," in *Proceedings of the 6th European Conference on Wireless Sensor Networks (EWSN '09)*, pp. 296–311, Cork, Ireland, February 2009.

[8] M. K. Khan and J. Zhang, "An efficient and practical fingerprint-based remote user authentication scheme with smart cards," in *Information Security Practice and Experience: Proceedings of the 2nd International Conference, ISPEC 2006, Hangzhou, China, April 11–14, 2006*, vol. 3903 of *Lecture Notes in Computer Science*, pp. 260–268, Springer, Berlin, Germany, 2006.

[9] E.-J. Yoon, M. K. Khan, and K.-Y. Yoo, "Cryptanalysis of a handover authentication scheme using credentials based on chameleon hashing," *IEICE Transactions on Information and Systems*, vol. E93-D, no. 12, pp. 3400–3402, 2010.

[10] S. Karthik, K. Vanitha, and G. Radhamani, "Trust management techniques in wireless sensor networks: an evaluation," in *Proceedings of the International Conference on Communications and Signal Processing (ICCSP '11)*, pp. 328–330, February 2011.

[11] F. G. Mármol and G. M. Pérez, "TRMSim-WSN, trust and reputation models simulator for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, IEEE Press, Piscataway, NJ, USA, June 2009.

[12] N. Karthik and V. R. S. Dhulipala, "Trust calculation in wireless sensor networks," in *Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT '11)*, vol. 4, pp. 376–380, Kanyakumari, India, April 2011.

[13] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proceedings of the 17th IEEE Symposium on Security and Privacy*, pp. 164–173, May 1996.

[14] A. Boukerch, L. Xu, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2413–2427, 2007.

[15] V. Oleshchuk, "Trust-based framework for security enhancement of wireless sensor networks," in *Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS '07)*, pp. 623–627, Dortmund, Germany, September 2007.

[16] V. R. S. Dhulipala, B. V. Prabha, and R. M. Chandrasekaran, "Trust worthy architecture for mobile Adhoc networks," in *Proceedings of the International Conference on Recent Trends in Business Administration and Information Processing (BAIP '10)*, vol. CCIS 70, pp. 557–560, Springer, Berlin, Germany, 2010.

[17] L. Abusalah, A. Khokhar, G. BenBrahim, and W. ElHajj, "TARP: trust-aware routing protocol," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '06)*, pp. 135–140, Vancouver, Canada, July 2006.

[18] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1698–1712, 2009.

[19] A. Raha, S. S. Babu, M. K. Naskar, O. Alfandi, and D. Hogrefe, "Trust integrated link state routing protocol for Wireless Sensor Networks (TILSRP)," in *Proceedings of the 5th IEEE International Conference on Advanced Networks and Telecommunication Systems (ANTS '11)*, pp. 1–6, Bangalore, India, December 2011.

[20] N. Poolsappasit and S. Madria, "A secure data aggregation based trust management approach for dealing with untrustworthy motes in sensor network," in *Proceedings of the 40th International Conference on Parallel Processing (ICPP '11)*, pp. 138–147, September 2011.

[21] G. Zhan, W. Shi, and J. Deng, "Design and implementation of TARF: a trust-aware routing framework for WSNs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 2, pp. 184–197, 2012.

[22] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–183, 2012.

[23] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[24] S. H. Jokhio, I. A. Jokhio, and A. H. Kemp, "Node capture attack detection and defence in wireless sensor networks," *IET Wireless Sensor Systems*, vol. 2, no. 3, pp. 161–169, 2012.

[25] J. Rodríguez-Molina, J.-F. Martínez, P. Castillejo, and L. López, "Combining wireless sensor networks and semantic middleware for an internet of things-based sportsman/woman monitoring application," *Sensors*, vol. 13, no. 2, pp. 1787–1835, 2013.